Web Programming DA

Exercise-15

Register Number: 23BCE1145

Name: S. Vishwajith

Question-1:

Q1.js

import React from "react";

function Header(*props*){

  return(

    <div>

      <header>

        <h1 *align*='center'>{*props*.title}</h1>

      </header>

    </div>

  );

}

function Jokes(){

  var Joke = ["What did one snowman say to the other snowman? It smells like carrots over here!", "Why did Beethoven get rid of his chickens? All they ever said was, “Bach, Bach, Bach!”", "What do you call a fake noodle? An impasta!", "Why does it take pirates a long time to learn the alphabet? Because they can spend years at C!"];

  return(

    <div *style*={{textAlign:"center"}}>

      <button *onClick*={() => {

        var num = Math.floor(Math.random() \* 4);

        document.getElementById("joke").innerHTML = Joke[num];

      }}>Generate a joke!</button>

      <p *id*="joke"></p>

    </div>

  );

}

function Footer(){

  return(

    <div *style*={{textAlign:"center"}}>

      <footer>

        <h2>This is a static footer</h2>

      </footer>

    </div>

  );

}

function Q1(){

  return (

    <div>

      <*Header* *title*="Generate Random Joke" />

      <*Jokes* />

      <*Footer* />

    </div>

  );

}

export default Q1;

### Output:

A screenshot of a computer

AI-generated content may be incorrect.

## Question-2:

### Q2.js

import React from "react";

function Q2(){

    return(

        <div *style*={{textAlign:'center', height:'100vh', display:'flex', justifyContent:'center', alignItems:'center'}}>

            <button *style*={{backgroundColor:'grey', color:'darkblue', padding:'10px', fontSize:'20px'}}>This is a styled button with inline CSS</button>

        </div>

    );

}

export default Q2;

### Output:

A screenshot of a computer

AI-generated content may be incorrect.

## Question-3:

### Q3.js

import React from "react";

function Q3(){

    var Button = {backgroundColor:'grey', color:'darkblue', padding:'10px', fontSize:'20px'};

    return(

        <div *style*={{textAlign:'center', height:'100vh', display:'flex', justifyContent:'center', alignItems:'center'}}>

            <button *style*={Button}>This is a styled button with internal CSS</button>

        </div>

    );

}

export default Q3;

### Output:

A screenshot of a computer

AI-generated content may be incorrect.

## Question-4:

### Q4.js

import React from "react";

import './Q4.css'

function Q4(){

    return(

        <div *style*={{textAlign:'center', height:'100vh', display:'flex', justifyContent:'center', alignItems:'center'}}>

            <button *className*="Button">This is a styled button with external css</button>

        </div>

    );

}

export default Q4;

### Q4.css

*.Button*{

    background-color:grey;

    color:darkblue;

    padding:10px;

    font-size:20px;

}

### Output: A screenshot of a computer AI-generated content may be incorrect.

## Question-5:

### Q5.js

import React, { useState } from "react";

import LifeCycleDemo from "./LifeCycleDemo";

function Q5() {

    const [showComponent, setShowComponent] = useState(true);

    return (

        <div *style*={{ textAlign: "center", fontSize: "20px" }}>

            <button *onClick*={() => setShowComponent(!showComponent)}>

                {showComponent ? "Unmount LifeCycleDemo" : "Mount LifeCycleDemo"}

            </button>

            {showComponent && <*LifeCycleDemo* />}

        </div>

    );

}

export default Q5;

### LifeCycleDemo.js

import React from "react";

class LifeCycleDemo extends *React*.Component{

    constructor(*props*){

*super*(*props*);

*this*.state = {

            count: 0,

            message: "Constructor was called - constructor()."

        };

        console.log("Constructor was called - constructor().");

    }

    componentDidMount(){

*this*.setState({message: "Component was mounted - componentDidMount()."});

        console.log("Component was mounted - componentDidMount().");

    }

    componentDidUpdate(*prevProps*, *prevState*){

        if(*this*.state.count !== *prevState*.count){

*this*.setState({message: "Component was updated - componentDidUpdate()."});

            console.log("Component was updated - componentDidUpdate().");

        }

    }

    componentWillUnmount(){

        console.log("Component will unmount - componentWillUnmount().");

    }

    render(){

        return(

            <div *style*={{textAlign:'center', fontSize: '20px'}}>

                <h1>Life Cycle Demo</h1>

                <p>Message: {*this*.state.message}</p>

                <button *onClick*={() => *this*.setState({count: *this*.state.count + 1})}>Increase</button> &emsp;

                <button *onClick*={() => *this*.setState({count: *this*.state.count - 1})}>Decrease</button> <br/>

                <p>Count: {*this*.state.count}</p>

            </div>

        );

    }

}

export default LifeCycleDemo;

### Output: A screenshot of a computer AI-generated content may be incorrect.A screenshot of a computer AI-generated content may be incorrect.A screenshot of a computer AI-generated content may be incorrect.

## Question-6:

### Q6.js

import React, {useState, useReducer} from "react";

const counterReducer = (*state*, *action*) => {

    switch(*action*.type){

        case 'increment':

            return { count: *state*.count + 1 };

        case 'decrement':

            if(*state*.count === 0){

                console.log("Count cannot be negative - userReducer.");

                return *state*;

            }

            else{

                return { count: *state*.count - 1 };

            }

        default:

            return *state*;

    }

};

function Decrease(*count*, *setCount*){

    if(*count* === 0){

        console.log("Count cannot be negative - useState.");

    }

    else{

        setCount(*count* - 1);

    }

}

function Q6(){

    const [count, setCount] = useState(0);

    const [counter, dispatchCounter] = useReducer(counterReducer, { count: 0 });

    var ButStyle = {fontSize: '15px', padding:'5px', margin:'5px'};

    return(

        <div *style*={{textAlign:'center', fontSize: '20px'}}>

            <button *style*={ButStyle} *onClick*={() => setCount(count + 1)}>Increase</button> &emsp;

            <button *style*={ButStyle} *onClick*={() => Decrease(count, setCount)}>Decrease</button> <br/>

            <p>Count using useState: {count}</p>

            <button *style*={ButStyle} *onClick*={() => dispatchCounter({ type: 'increment' })}>Increase</button> &emsp;

            <button *style*={ButStyle} *onClick*={() => dispatchCounter({ type: 'decrement' })}>Decrease</button> <br/>

            <p>Count using useReducer: {counter.count}</p>

        </div>

    );

}

export default Q6;

### Output: A screenshot of a computer AI-generated content may be incorrect.

## Question-7:

### Q7.js

import React, {useState, useEffect} from "react";

function Q7(){

    const [joke, setJoke] = useState("");

    const [loading, setLoading] = useState(true);

    const [error, setError] = useState(null);

    const [click, setClick] = useState(false);

    useEffect(() => {

        fetch("https://api.chucknorris.io/jokes/random")

            .then((*response*) => {

                if (!*response*.ok) {

                    throw **new** *Error*("Network response was not ok");

                }

                return *response*.json();

            })

            .then((*data*) => {

                setJoke(*data*.value);

                setLoading(false);

            })

            .catch((*error*) => {

                setError(*error*);

                setLoading(false);

            });

    }, [click]);

    return(

        <div *style*={{textAlign:'center', fontSize: '20px'}}>

            <h1>Random Joke from API</h1>

            {loading && <p>Loading...</p>}

            {error && <p>Error: {error.message}</p>}

            {!loading && !error && <p>{joke}</p>}

            <button *onClick*={() => setClick(!click)}>Fetch Another Joke</button>

        </div>

    )

}

export default Q7;

### Output: A screenshot of a computer AI-generated content may be incorrect.

## Question-8:

### Q8.js

import React, {useRef} from "react";

function Q8(){

    const inputRef = useRef(null);

    const handleClick = () => {

        inputRef.current.focus();

    };

    return (

        <div *style*={{ textAlign: "center", display: "flex", alignItems: "center", justifyContent: "center", height: "100vh" }}>

            <input *type*="text" *ref*={inputRef} *placeholder*="Focus on this." />

            <button *onClick*={handleClick}>Focus Input</button>

        </div>

    );

}

export default Q8;

### Output: A screenshot of a computer AI-generated content may be incorrect.A screenshot of a computer AI-generated content may be incorrect.

## Question-9:

### Q9.js

import React, { createContext, useContext, useState } from "react";

const ThemeContext = createContext();

const useTheme = () => useContext(ThemeContext);

const ThemeProvider = ({ *children* }) => {

  const [theme, setTheme] = useState("light");

  const toggleTheme = () => {

    setTheme((*prev*) => (*prev* === "light" ? "dark" : "light"));

  };

  return (

    <*ThemeContext.Provider* *value*={{ theme, toggleTheme }}>

      {*children*}

    </*ThemeContext.Provider*>

  );

};

const ThemeToggle = () => {

  const { theme, toggleTheme } = useTheme();

  return (

    <div

*style*={{

        textAlign: "center",

        marginTop: "20px",

      }}

    >

      <button

*onClick*={toggleTheme}

*style*={{

          padding: "10px 20px",

          fontSize: "18px",

          cursor: "pointer",

          backgroundColor: theme === "light" ? "#333" : "#fff",

          color: theme === "light" ? "#fff" : "#333",

          border: "none",

          borderRadius: "5px",

        }}

      >

        Switch to {theme === "light" ? "Dark" : "Light"} Mode

      </button>

    </div>

  );

};

const Header = () => {

  const { theme } = useTheme();

  return (

    <header

*style*={{

        backgroundColor: theme === "light" ? "#4CAF50" : "#555",

        padding: "10px",

        textAlign: "center",

        color: "#fff",

      }}

    >

      <h1>{theme === "light" ? "Light Theme Activated." : "Dark Theme Activated."}</h1>

    </header>

  );

};

const Content = () => {

  const { theme } = useTheme();

  return (

    <div

*style*={{

        padding: "20px",

        backgroundColor: theme === "light" ? "#f5f5f5" : "#333",

        color: theme === "light" ? "#333" : "#fff",

        transition: "0.3s ease-in-out",

        height: "63.5vh",

        display: "flex",

        justifyContent: "center",

        alignItems: "center",

      }}

    >

      <p *style*={{ fontSize: "20px" }}>

        {theme === "light"

          ? "Enjoy the bright vibes!"

          : "Darkness is where real code thrives!"}

      </p>

    </div>

  );

};

const Footer = () => {

  const { theme } = useTheme();

  return (

    <footer

*style*={{

        backgroundColor: theme === "light" ? "#4CAF50" : "#555",

        padding: "10px",

        textAlign: "center",

        color: "#fff",

      }}

    >

      <p>Theme Toggle App - Q9</p>

    </footer>

  );

};

const Q9 = () => {

  return (

    <*ThemeProvider*>

      <*Header* />

      <*ThemeToggle* />

      <*Content* />

      <*Footer* />

    </*ThemeProvider*>

  );

};

export default Q9;

### Output: A screenshot of a computer AI-generated content may be incorrect.A screenshot of a computer AI-generated content may be incorrect.

## Question-10:

### Q10.js

import React from "react";

function Child(*props*){

    return(

        <div>

            <h2>Child Component</h2>

            <p>Message from Parent: "{*props*.message}"</p>

        </div>

    );

}

function Q10(){

    const message = "Hello from Parent Component!";

    return (

        <div *style*={{ textAlign: "center", marginTop: "20px" }}>

            <h1>Parent Component</h1>

            <p>Message to Child: "{message}"</p>

            <*Child* *message*={message} />

        </div>

    );

}

export default Q10;

### Output: A screenshot of a computer AI-generated content may be incorrect.

## Question-11:

### Q11.js

import React from "react";

import PropTypes from "prop-types";

function Child(*props*){

    return(

        <div>

            <h2>Child Component</h2>

            <p>Message from Parent: "{*props*.message}"</p>

        </div>

    );

}

function Q11(){

    const message = "Hello from Parent Component!";

    return (

        <div *style*={{ textAlign: "center", marginTop: "20px" }}>

            <h1>Parent Component</h1>

            <p>Message to Child: "{message}"</p>

            <*Child* *message*={message} />

        </div>

    );

}

Q11.propTypes = {

    message: *PropTypes*.string.isRequired,

}

export default Q11;

### Output: A screenshot of a computer AI-generated content may be incorrect.

## Question-12:

import React from "react";

function Form1(){

    const [name, setName] = React.useState("");

    const [email, setEmail] = React.useState("");

    const handleSubmit1 = (*e*) => {

*e*.preventDefault();

        console.log("Form 1 pressed submit.");

    }

    return(

        <div *style*={{ textAlign: "center", marginTop: "20px" }}>

            <h1>Form 1</h1>

            <form>

                <label>Name:</label><br/>

                <input *type*="text" *value*={name} *onChange*={(*e*) => setName(*e*.target.value)} /><br/><br/>

                <label>Email:</label><br/>

                <input *type*="email" *value*={email} *onChange*={(*e*) => setEmail(*e*.target.value)} /><br/><br/>

                <button *type*="submit" *onClick*={handleSubmit1}>Submit</button>

            </form>

            <p>Name: {name}</p>

            <p>Email: {email}</p>

        </div>

    );

}

function Form2(){

    const inputName = *React*.useRef(null);

    const inputEmail = *React*.useRef(null);

    const handleSubmit2 = (*e*) => {

*e*.preventDefault();

        const name = inputName.current.value;

        const email = inputEmail.current.value;

        document.getElementById("name").innerText = `Name: ${name}`;

        document.getElementById("email").innerText = `Email: ${email}`;

    }

    return(

        <div *style*={{ textAlign: "center", marginTop: "20px" }}>

            <h1>Form 2</h1>

            <form>

                <label>Name:</label><br/>

                <input *type*="text" *ref*={inputName} /><br/><br/>

                <label>Email:</label><br/>

                <input *type*="email" *ref*={inputEmail} /><br/><br/>

                <button *type*="submit" *onClick*={handleSubmit2}>Submit</button>

            </form>

            <p *id*="name"></p>

            <p *id*="email"></p>

        </div>

    );

}

function Q12(){

    return (

        <div>

            <*Form1* />

            <*Form2* />

        </div>

    );

}

export default Q12;

### Output: A screenshot of a computer AI-generated content may be incorrect.A screenshot of a computer AI-generated content may be incorrect.